

RESEARCH

Open Access

Dynamic social cloud management scheme based on transformable Stackelberg game



Sungwook Kim

Abstract

With the ubiquitous nature of social networks and cloud computing, we are starting to explore a new way to interact with and exploit these developing paradigms. Social cloud (SC) is a service or resource sharing framework on top of social networks and built on the trust-based social relationships. In recent years, the idea of SC has been gaining importance because of its potential applicability. This article introduces a novel SC management scheme with a view of game theory model and reciprocal resource sharing mechanism. In particular, we devise a new transformable Stackelberg game to coordinate the interdependence between social structure and resource availability for individual users. Our proposed scheme constantly monitors the current SC system conditions and adaptively exploits the available resources while ensuring mutual fairness. The simulation results show that the proposed method is effective in distributed SC environments and adaptively supports application executions timely and ubiquitously.

Keywords: Social cloud system, Transformable Stackelberg game, Dynamic resource sharing, Reciprocal fairness, Game theory

1 Introduction

Digital relationships between individual people become more and more embedded in our daily actions, and they can be powerful influences in our real-life. Moreover, we are now connected with all our social networks through mobile devices. The increasing ubiquity of social networks is evidenced by the growing popularity of social network services. A social network service consists of a representation of each user, his or her social links, and a variety of additional services. Usually, social networks provide a platform to facilitate communications and resource sharing between users while modeling real-world relationships. Therefore, a variety of social network services have extended beyond simple communication among users [1–3].

With the advent of social networks, cloud computing is becoming an emerging paradigm to provide a flexible stack of computing, software, and storage services. In a scalable and virtualized manner over networks, cloud users can access to fully virtualized hardware resources. The adoption of cloud computing technology is attractive; users obtain cloud resources, whose management is partly

automated and can be scaled almost instantaneously. However, with the rapid development of cloud computing, critical issues of cloud computing technology have emerged. In general, modern cloud applications are characterized by assuming a constant environment. But real-world environments are open, dynamic, and unpredictable [4–7].

In social networks, individual users are bound by finite resource capacity and limited capabilities. However, some users may have surplus resource capacity or capabilities. Therefore, the superfluous resource could be shared for a mutual benefit. Within the context of a social network, users may wish to share resources without payment and utilize a reciprocal credit based on the trust model [8, 9]. To satisfy this goal, a new concept, social cloud (SC) was introduced by combining the methodologies of social networks and cloud computing. SC is a novel scalable computing model where resources are beneficially shared among a group of social network users. From [8], we rehearse the formal definition of SC as follows: *A social cloud is a resource and service sharing framework utilizing relationships established between members of a social network.* Based on the cloud

Correspondence: swkim01@sogang.ac.kr
Department of Computer Science, Sogang University, 35 Baekbeom-ro (Sinsu-dong), Mapo-gu, Seoul 121-742, South Korea

computing technique, SC model is used to enable virtualized resource sharing through service-based interfaces [8].

To construct the SC system in a real-world environment, there are many challenges that need to be carefully considered. First of all, the concept of SC focuses on the sharing rather than sale of resources. Using sharing preferences, the social context of exchange is accentuated along with the social ties of individual users [10]. However, social relationships are not simply edges in a graph. There are many different types of relationship; different users will associate different levels of trust to different relationship contexts and have different reliability, trustworthiness, and availability. Therefore, users may have very specific preferences with whom they interact. To design an effective SC control scheme, it is necessary to take into account the preferences and perceptions of users toward one another [10].

Under widely dynamic SC system conditions, end users can be assumed as intelligent rational decision-makers, and they select a best-response strategy to maximize their expected payoffs. This situation is well-suited for the game theory. Game theory is a field of applied mathematics that provides an effective tool to model interactions among independent decision-makers. It can describe the reactions of one set of decision-makers to another and analyze the situations in terms of conflict and cooperation. Therefore, game theory is really useful in analyzing the mutual interactions among multi-users. Thus, it can be a major paradigm to retain an equilibrium between different users that feature complex interactive relations [11].

In 1934, German economist H. V. Stackelberg proposed a hierarchical strategic game model based on two kinds of different decision-makers. Under a hierarchical decision-making structure, one or more players declare and announce their strategies before the other players choose their strategies. In game theory terms, the declaring players are called as leaders while the players who react to the leaders are called as followers. Originally, the Stackelberg game model was developed to explain the monopoly of industry. The leader is the incumbent monopoly of the industry, and the follower is a new entrant; it can be the static bilevel optimization model [11]. In this study, we have further extended the classical Stackelberg model and developed a novel game mode, called transformable Stackelberg (TS) game model. In the TS game, each player can be a leader or a follower as the case may be. Therefore, the position of game players is dynamically transformable according to current conditions.

Motivated by the above discussion, we propose a new SC resource sharing scheme based on the TS game model. TS game model is a useful framework for designing decentralized mechanisms, such that users in SC systems can self-organize into the mutually satisfactory resource

sharing process. This self-organizing feature can add autonomies into SC systems and help to ease the heavy burden of complex centralized control algorithms. Especially, we pay serious attention to trust evaluation, repeated interactions, and iterative self-learning techniques to effectively implement our resource sharing process. In the proposed scheme, such techniques have been incorporated into the TS game model and work together toward an effective system performance. Therefore, we can induce all users to share their resources adaptively. The major contributions of the proposed scheme are (i) the adjustable dynamics considering the current SC environments, (ii) the interactive learning process based on the iterative feedback mechanism, (iii) the sophisticated combination of the reciprocal relationship and incentive mechanism, and (iv) practical approach to effectively reach a desirable solution. Other existing schemes [8, 10, 12–16] cannot offer these attractive features.

1.1 Related work

The area of numerical methods or algorithms for efficient SC control problems has been extensively studied and has received considerable attention in recent years [8, 10, 12–16]. In [14], Lee et al. presented the design and development of the architecture of the just-in-time social cloud and experimental methods to measure the just-in-time social influences. Just-in-time social cloud service can better guide people toward their long-term goals by influencing their choices at the moment and potentially mitigating behavioral biases. They attempted to design a social platform that can be programmed to benefit human being's long-term goals by mitigating the inter-temporal biases people have toward present [14]. The scheme in [8] was developed for leveraging the online relationships to form a dynamic social cloud, while enabling users to share heterogeneous resources within the context of a social network. The socially corrective mechanism was used to enable a cloud-based framework for long term sharing with lower privacy concerns and security overheads that were present in traditional cloud environments [8].

Ali et al. developed the Cloud Resource Bartering (CRB) model for sharing user's computational resources through a social network [15]. The CRB model allowed users of online social network to share their cloud resources without money changing hands. This scheme linked a social network with the computational cloud to create a social cloud so that users can share their part of the cloud with their social community [15]. The scheme in [16] considered the effect of reputation when parties interacted in social cloud to find a new way realizing mutual cooperation. Parties in the social cloud were rational who valued their reputation. Cooperation can boost their reputation, so they had incentives to cooperate with

others such that they may get a higher utility. The basic idea in [16] was to add reputation deriving from social cloud as part of the utility.

The *Social Compute Cloud (SCC)* scheme [10] was developed for the SC interaction system. This scheme has presented a social compute cloud platform that enabled the sharing of infrastructure resources between friends via digitally encoded social relationships. To construct a social compute cloud, the *SCC* scheme accessed users' social networks, allowed users to elicit sharing preferences, and utilized matching algorithms to enable preference-based socially aware resource allocation. The *Incentive-based Social Cloud (ISC)* scheme [12] was designed to model the selfish behavior of the users who were supplying resources and aiming to maximize their own benefits. Based on the reputation-based pricing and collective punishment mechanism, the *ISC* scheme compelled suppliers to change their selfish strategies in a manner that improved the efficiency of the SC system.

The *Reputation-based Social Cloud (RSC)* scheme [13] added the concept of reputations as part of the utility. This scheme described the architecture and interaction between two rational parties in the social cloud, where two parties received their opponent's trust or reputation from the social cloud. In the *RSC* scheme, the reputation was affected by the interactions with other parties in the social network. As mentioned above, numerous studies have shown how social networks create social influences on people's choices across time and space. In this study, we compared the performance of the proposed scheme with the existing schemes in [10, 12] and [13] to confirm the superiority of our approach.

This paper is organized as follows. Section 2 describes how insights from TS game model would help us to incorporate learning mechanism into the SC control scheme and to guide selfish users to achieve a globally desirable SC system performance. Afterwards, the main step of the proposed SC control algorithm is presented. In Section 3, performance evaluation results are presented along with comparisons with the schemes proposed in [10, 12] and [13]. Through simulation, we show the ability of proposed scheme to achieve high accuracy and promptness in dynamic SC environments. Finally, we end up with some concluding remarks in Section 4.

2 Proposed social cloud resource sharing algorithms

In this section, we introduce the proposed SC resource sharing scheme in detail. Based on the TS game model, our proposed scheme can approximate a globally desirable system performance while ensuring user cooperations.

2.1 Transformable Stackelberg game model

Social cloud is a form of community cloud and is designed to enable access to elastic compute capabilities contributed by socially connected community [10]. To avoid the social dilemma such as "Tragedy of the Commons", social incentives motivate users to participate in, and contribute to, SC systems in different ways. Motivation is generally categorized as either intrinsic or extrinsic. Extrinsic motivation represents that users are motivated by an external reward, e.g., virtual currency. Therefore, they will contribute to the SC while the expected benefits exceed the cost of contribution. Intrinsic motivation represents an internal satisfaction obtained from the task itself rather than the rewards or benefits. In realities, people incline to cooperate with others for reciprocation and altruism. These factors rationalize non-economic behaviors and motivates users to contribute to SC [8].

In this study, we leverage social incentives to create ad hoc clouds without incurring the overhead of central complex processes. To implement a fair-efficient resource sharing mechanism, we pay serious attention to contribution evaluation, repeated interactions, and iterative self-learning algorithms. In the proposed scheme, these techniques have been incorporated into the TS game model, which is developed to let distributed players learn the best strategy in the step-by-step interactive online manner. This approach can induce all SC users to share resources as much as possible and ensure a good tradeoff between the implementation complexity for real-world SC operations and an effective system performance. Therefore, the proposed scheme can be used to overcome one of the major limitations of traditional SC monitoring methods.

To characterize our proposed scheme, we design a new TS game model \mathbb{G} for SC systems. In a realistic SC scenario, each user, i.e., network device, can be a resource supplier or demander. Suppliers make their decisions by considering the possible reactions of demanders. Demanders react dependently based on the decision of suppliers while attempting to maximize their satisfaction. Therefore, in our TS game model \mathbb{G} , suppliers plays the role of leaders and demanders become followers. Based on these assumptions, \mathbb{G} is defined as a tuple $\mathbb{G} = (\mathbb{N}, (V_i)_{i \in \mathbb{N}}, (S_i)_{i \in \mathbb{N}}, (A_i)_{i \in \mathbb{N}}, (U_i)_{i \in \mathbb{N}}, T)$ at each time period t of gameplay.

- \mathbb{N} is the finite set of players, and $\mathbb{N} = \{p_1, \dots, p_n\}$ where $p_{i, 1 \leq i \leq n}$ represents the i th user. A player can be a supplier or a demander at times. Therefore, the position of each player would be dynamically changeable as a leader or a follower.

- V_i is the amount of exchangeable resources of the player i . In this study, V is the computing capacity, e.g., CPU cycles.
- S_i is the set of strategies with the player i . If the player i is a supplier, S_i can be defined as the amount of sharing resource. If the player i is a demander, S_i is defined as the amount of requested resource.
- Λ_i is the contribution level of the player i in the SC community.
- The U_i is the payoff received by the player i . Traditionally, the payoff is determined as the obtained outcome minus the cost to obtain that outcome.
- The T is a time period. The \mathbb{G} is repeated $t \in T < \infty$ time periods with competitive and cooperative manner.

In the SC system, each network device has its own computation resources and executes elastic applications. Applications can be divided into two parts: one part runs locally and the other part can be executed on the cloud side. Therefore, applications in each network device can be computed either locally or remotely via computation offloading. In general, the main challenges to design an offloading mechanism are to decide what, when, and how to be offloaded. In the proposed scheme, available resources in suppliers are matched to demanders based on the supplier-demander interactive relationship. According to the proposed TS game model, network devices can self-organize into the mutually satisfactory computation offloading decisions.

2.2 Resource sharing process in social cloud systems

Different users may pursue individually to maximize their profits. From the viewpoint of demanders, the payoff corresponds to the resource sharing benefit minus the incurred cost to share the remote resource. Therefore, the utility function of demander i (U_i^D) is defined as follows:

$$U_i^D(x_i, \Lambda_i) = \mathcal{B}_i(j, x_i) - C_i(j, x_i), \quad j \text{ is a supplier } \in N \text{ and } i \neq j \tag{1}$$

where \mathcal{X}_i is the requested resource amount, and $\mathcal{B}_i(\cdot)$ and $C_i(\cdot)$ are the benefit and cost functions for the demander i . Usually, elastic applications have concave benefit function, which provides monotone increasing values in proportion to the assigned resource amounts. According to the amount of assigned resource, $\mathcal{B}_i(\cdot)$ and $C_i(\cdot)$ are given by

$$\begin{aligned} \mathcal{B}_i(j, x_i) &= \sin\left(\frac{\pi}{2} \times \frac{b_j^i}{x_i}\right) \text{ and } C_i(j, x_i) = \zeta \times \left(\varrho \times \frac{b_j^i}{\mathcal{M}\mathcal{X}}\right) \\ \text{s.t., } \zeta &= b_j^i / \max\{\Lambda_i, b_j^i\} \quad \text{and } \varrho = \mathcal{E}(b_j^i) / \mathcal{E}(\mathcal{M}\mathcal{X}) \end{aligned} \tag{2}$$

where b_j^i is the assigned resource amount from the supplier j . ζ is a cost control parameter, and $\mathcal{M}\mathcal{X}$ is the total resource amount to process the corresponding application. $\mathcal{E}(\mathcal{M}\mathcal{X})$ and $\mathcal{E}(b_j^i)$ are the energy consumption to execute $\mathcal{M}\mathcal{X}$ and b_j^i amount resources, respectively. In this study, $\mathcal{E}(\cdot)$ is a linear function. Λ_i is the accumulated contributiveness of the demander i . After the remote execution, Λ_i is decreased by b_j^i , i.e., $\Lambda_i = \Lambda_i - b_j^i$. Based on the expected payoff $U^D(\cdot)$, demanders can try to find the best actions, i.e., the decision of \mathcal{X}_i amount.

From the viewpoint of suppliers, the payoff also corresponds to the received benefit minus the incurred cost to assign the sharing resource. However, in contrast to the demanders' interest, the sharing benefit is defined according to the reciprocal cooperation, more generally, the combination of evolution, altruism, and reciprocity. In this study, we assume that users can be altruistic toward others and react to other users' altruism. Therefore, the received benefit function is developed based on the simple reciprocal mechanism. By considering the service cost, the supplier j 's utility function to the demander i ($U_j^S(\cdot)$) is defined as follows:

$$U_j^S(\mathcal{Z}_j, \Lambda_j, i) = \mathbb{B}_j(\mathcal{Z}_j, \Lambda_j, i) - C_j(\mathcal{Z}_j) \tag{3}$$

where \mathcal{Z}_j is the amount of sharing resource of the supplier j and $\mathbb{B}_j(\mathcal{Z}_j, \Lambda_j, i)$ and $C_j(\mathcal{Z}_j)$ are the benefit and cost functions for the supplier j , respectively. To get the optimal payoff, suppliers try to maximize their benefit function while minimizing their cost function. According to the \mathcal{Z}_j and Λ values, $\mathbb{B}_j(\cdot)$ and $C_j(\cdot)$ are given by

$$\begin{aligned} \mathbb{B}_j(\mathcal{Z}_j, \Lambda_j, i) &= \left[\left(\theta_j^i \times e^{\mathcal{Z}_j} \right) + \mathcal{F}_j(\Lambda_j) \right] \text{ and } C_j(\mathcal{Z}_j) \\ &= \lambda \times \left(\mathcal{E}(\mathcal{Z}_j) / \mathcal{E}(\mathcal{I}) \right) \\ \text{s.t., } \theta_j^i &= \frac{\phi_j + \left(\phi_j \times \left(\Lambda_i / (\Lambda_i + \Lambda_j) \right) \right)}{1 - \left(\Lambda_i / (\Lambda_i + \Lambda_j) \right)} \text{ and } \mathcal{F}_j(\Lambda_j) \\ &= \left(\mathcal{Z}_j / \max\{\Lambda_j, \mathcal{Z}_j\} \right) \end{aligned} \tag{4}$$

θ_j^i is the supplier j 's altruistic parameter to the demander i , and ϕ_j is the supplier j 's general altruistic propensity. $\mathcal{E}(\mathcal{I})$ and $\mathcal{E}(\mathcal{Z}_j)$ are the energy consumption to execute

the supplier j 's total resource (\mathfrak{T}) and \mathcal{Z}_j , respectively. λ is the cost control parameter. After the resource sharing process, Λ_j is increased by \mathcal{Z}_j , i.e., $\Lambda_j = \Lambda_j + \mathcal{Z}_j$.

Under dynamically changing SC environments, a fixed altruistic propensity cannot effectively adapt to the current SC condition. Therefore, the ϕ value should be dynamically adjustable. In order to implement the ϕ value adjustment process, suppliers should learn how to perform well by interacting with demanders and dynamically adjust their ϕ levels. Based on the exponential weight learning algorithm [17], suppliers in our TS game model can constantly adapt each ϕ level to get an appropriate attitude to their corresponding SC environments. Let \mathbb{K} be the set of all possible altruistic propensity levels, i.e., $\phi \in \mathbb{K}$. In the proposed learning algorithm, the probability of choosing the k 's propensity level in \mathbb{K} at time t ($P_k^\phi(t)$) is defined by

$$P_k^\phi(t) = (1-\gamma) \times \left(\frac{\omega_k(t)}{\sum_{j=1}^K \omega_j(t)} \right) + \frac{\gamma}{\|\mathbb{K}\|}$$

$$\text{s.t., } \omega_j(t) = \omega_j(t-1) \times \exp\left(\gamma \times \left[\frac{U_j(t-1)}{(P_j^\phi(t-1) \times \|\mathbb{K}\|)} \right]\right)$$
(5)

where $\gamma \in [0,1]$ is an egalitarianism factor, which tunes the desire to pick an action uniformly at random. That is, if $\gamma = 1$, the weights have no effect on the choices at any step. $\|\mathbb{K}\|$ is the total number of propensity levels, and $U_j(t-1)$ is the obtained payoff ($U_j^S(\cdot)$) at time $t-1$. According to the distribution of $P(t)$, suppliers can modify their ϕ levels without any impractical rationality assumptions. During the step-by-step iteration, suppliers individually adjust the ϕ value by using the dynamics of feedback-based repeated process. Therefore, under dynamic SC situations, the main advantage of our proposed approach is a real-world practicality.

During real-world SC operations, multiple demanders can request the resource sharing from the same supplier. In this case, the role of supplier is to distribute dynamically the limited resource for each demander. To get a fair-efficient resource allocation, we develop a new resource distribution algorithm based on the relative utilitarian bargaining model [18]; it can be applicable and useful in a SC system with a frequently changing situation. In the proposed scheme, we consider each demanders' Λ values as asymmetric bargaining powers. Therefore, our bargaining solution (\mathcal{R}_B) for resource distribution is given by

$$\mathcal{R}_B = \max_{b_j^i, i \in N_j} \left(\sum_{i \in N_j} \mathfrak{U}_i(b_j^i, \Lambda_i) \right), \quad \text{s.t., } \mathfrak{U}_i(b_j^i, \Lambda_i)$$

$$= \left(\frac{b_j^i}{x_i} \right)^{\eta_i} \text{ and } \eta_i = \Lambda_i / \sum_{j=1}^K \Lambda_j$$
(6)

where N_j is the set of all resource requesting demanders to the supplier j . \mathcal{R}_B is a vector, which corresponds to the resource distribution amounts to each demander.

In general, traditional game models have focused on investigating *which* decisions are made or *what* decisions should be made. Therefore, an equilibrium point is a well-known solution concept in classical game models. The strategy in equilibrium is the best response to the strategies of the other users. In the proposed TS game model, an equilibrium point of suppliers and demanders can be defined as follows:

$$U^* (U^{D*}, U^{S*}) = \begin{cases} U^{S*} = \begin{cases} \arg \max_{z_j \in \mathcal{S}_j} \{U_j^S(z_j, \Lambda_j, t)\}, & \text{if } j \text{ is a supplier} \\ & \text{with single demander } i \\ \arg \max_{i \in \mathcal{N}, z_j^i \in \mathcal{S}_j} \{U_j^S(z_j, \Lambda_j, \mathcal{N})\}, & \text{if } j \text{ is a supplier} \\ & \text{with multiple demanders } \mathcal{N} \end{cases} \\ U^{D*} = \arg \max_{x_i \in \mathcal{S}_i} \{U_i^D(x_i, \Lambda_i)\}, & \text{if } i \text{ is a demander} \end{cases}$$
(7)

In recent decades, there had been many conceptual and empirical critiques toward the equilibrium concept. First, in the scenario of equilibrium, the players are assumed to be fully rational. This perfect rational assumption requires complete information; all factors of the game should be common knowledge. However, in reality, this assumption is actually disputable and rarely holds. In particular, the hypothesis of exact rationality does not apply to many interactive situations. Second, the idea of equilibrium has mostly been developed in a static setting. Under the dynamic changing SC environments, it cannot capture the adaptation of players to change their strategies and reach equilibrium over time.

In this study, we introduce a new solution concept for the TS game model; it is the obtained consensus with reciprocal advantage. Such a consensus in multi-player decision-making process is defined as *Cooperative Consensus Equilibrium (CCE)*. During TS game operations, game players may adjust their altruistic propensities when outcome contradicts their beliefs and adaptively modify their altruistic propensities in an attempt to reach a mutually acceptable decision vector. Therefore, the solution concept of *CCE* presents a dynamic learning interpretation to adapt the current SC situations. To the best of our knowledge, this is the first study to address

the iterative SC resource sharing problem with a reciprocal consensus solution concept.

Definition *CCE* is a system status that can be obtained through repeating the TS game with receiving feedbacks. When all the accumulated contributiveness (Λ) of users are balanced, i.e., the relative contribution differences of users are less than a pre-defined maximum bound (Γ_Λ), this state is defined as the CCE. That is formally formulated as

$$\max_i \left\{ \left| \frac{(A_i / T^{i^M})}{\left(\sum_{k \in \mathbb{N}} A_k / T^{k^M} \right)} \right| \right\} < \Gamma_\Lambda \tag{8}$$

where T_k^M is the maximum resource capacity of user k 's device. Therefore, the main idea of *CCE* is to minimize the maximum unbalanced behavior degree of users.

2.3 The main steps of proposed algorithm

In recent years, SC has captured the attention of organizations by offering elastic, scalable, and cost-effective network resource management. In this study, we present a new TS game model for the interaction of multiple users with elastic applications. In the proposed TS game, a sophisticated combination of the reciprocal relationship and incentive mechanism can provide much more suitable resource sharing algorithm. Based on the real-time interactive feedback process, each user can adapt its behavior and act strategically to achieve a better profit. Usually, the traditional optimal solutions need exponential time complexity. However, the proposed solution concept only needs polynomial time complexity. Therefore, our approach can offer a more realistic model for multiple users with limited rationality. The proposed algorithm is described by the following major steps.

Step 1: At the start, all Λ values are set to the relatively same initial values, e.g., zero, and each altruistic propensity ϕ is randomly chosen from \mathbb{K} . When reciprocal interaction history is unavailable, it is a proper initialization. Control parameters, i.e., γ , λ and Γ_Λ , are listed in Table 1.

Step 2: When an individual device needs an additional resource, it becomes a demander and asks the x amount resource to maximize the expected payoff $U^D(\cdot)$ according to Eq. (1).

Step 3: If the neighboring nodes of a demander have enough available resources, they can be suppliers. Suppliers provide the Z amount resource to maximize the expected payoff $U^S(\cdot)$ according to Eq. (3). When multiple demanders request the resource simultaneously, a supplier distributes the available resource using the Eq. (6).

Table 1 Application and system parameters used in the simulation experiment

Application type	Total computation requirement ($\mathcal{M}\mathcal{X}$)	Minimum local computation requirement	Service duration average/second
I	1.28 Mbps	0.64 Mbps	1,800 s (30 min)
II	2.56 Mbps	1.28 Mbps	1,800 s (30 min)
III	3.84 Mbps	1.92 Mbps	600 s (5 min)
IV	5.12 Mbps	2.56 Mbps	600 s (5 min)
V	6.40 Mbps	3.20 Mbps	1,800 s (30 min)
VI	7.24 Mbps	3.62 Mbps	1,800 s (30 min)
VII	8.12 Mbps	4.06 Mbps	3,000 s (50 min)
VIII	9.48 Mbps	4.74 Mbps	1,200 s (20 min)
Parameter	Value	Description	
n	30	The number of mobile devices in SC	
ϕ	0.1, 0.2, 0.3, 0.4	Altruistic propensity levels for each user	
λ	1	The cost control parameter for a demander	
$ \mathbb{K} $	5	The number of altruistic propensity levels	
γ	0.3	The egalitarianism factor for learning algorithm	
Γ_Λ	$0.2 \leq \Gamma_\Lambda \leq 0.8$	A pre-defined maximum relative contribution bound	
Parameter	Description		
Λ_i	The contribution level of the player i in the SC community		
\mathcal{X}_i	The requested resource amount of the demander i		
ζ	A cost control parameter		
$\mathcal{M}\mathcal{X}$	The total resource amount to process the corresponding application		
\mathcal{Z}_j	The amount of sharing resource of the supplier j		
θ_j^i	The supplier j 's altruistic parameter to the demander i		
φ_j	The supplier j 's general altruistic propensity		

Step 4: Using the simple two-sided matching algorithm, a demander selects the most adaptable supplier, and the resource is effectively shared. After the resource sharing process, Λ is adjusted dynamically.

Step 5: In each game stage game, ϕ of each mobile device is periodically modified according to the exponential weight learning algorithm. Based on the adjusted $P_{k,k \in \mathbb{K}}^\phi$ in Eq. (5), an actual ϕ value of each device is selected stochastically.

Step 6: As game players, mobile devices are interrelated with elastic applications, and interact with each other in the TS game. Under widely diverse SC environments, this iterative feedback procedure continues to reach the *CCE* status.

Step 7: Mobile devices self-monitors the current SC situation in a distributed online manner; the next iteration resumes at step 2.

3 Performance evaluation

In this section, we compare the performance of the proposed scheme with other existing schemes [10, 12, 13] and confirm the performance superiority of the proposed approach using a simulation model. Our simulation model is a representation of an SC system that includes social network users and the behavior and interactions of these users. In order to ensure that the model is sufficiently generic to be valid in the real-world, the assumptions used in our simulation model were as follows:

- The simulated system consisted of 30 social network users for the SC platform.
- We followed the traditional network topology, decentralized, and partially connected topology.
- In the network coverage area, network mobile devices executed elastic applications. For each device, application service request was Poisson with rate ρ (services/s) and the range of the offered service load was varied from 0 to 3.0.
- Each mobile device has a different resource amount. There were five resource capacities for the devices and the device i 's capacity ($\mathfrak{C}_{i,i} \in \mathbb{N}$) was $\mathfrak{C}_i \in \{10, 12.5, 15, 17.5, \text{ and } 20 \text{ Mbps}\}$.
- To reduce the computation complexity, the amount of resource allocation is specified in terms of basic units (BUs), where one BU is the minimum amount (e.g., 320 kbps in our system) for the resource sharing process.
- The service durations of applications are exponentially distributed with different means for different multimedia application types.

- Network performance measures obtained based on 100 simulation runs were plotted as a function of the offered service load.
- The performance criteria obtained through simulation were resource usability, CCE convergence ratio, and normalized system throughput.

Table 1 shows the system parameters used in the simulation. In order to emulate a real SC system and perform a fair comparison, we used the system parameters for a realistic simulation model.

In this paper, we compared the performance of the proposed scheme with existing schemes: the SCC scheme [10], ISC scheme [12], and RSC scheme [13]. These existing schemes were recently developed as effective SC management algorithms; all the schemes have polynomial time complexity. Compared to these schemes, we can confirm the superiority of our proposed approach.

Figure 1 presents the performance comparison of each scheme in terms of resource usability in the SC systems. In this study, resource usability is a measure of how system resources are actually used. Traditionally, monitoring how resources are used is one of the most critical aspects of SC management. During the SC system operations, all schemes produced similar resource usability. However, the suppliers in our proposed scheme adaptively distribute resources to the demanders while considering reciprocal fairness. Therefore, the resource usability produced by the proposed scheme was higher than the other schemes from low to heavy service load intensities.

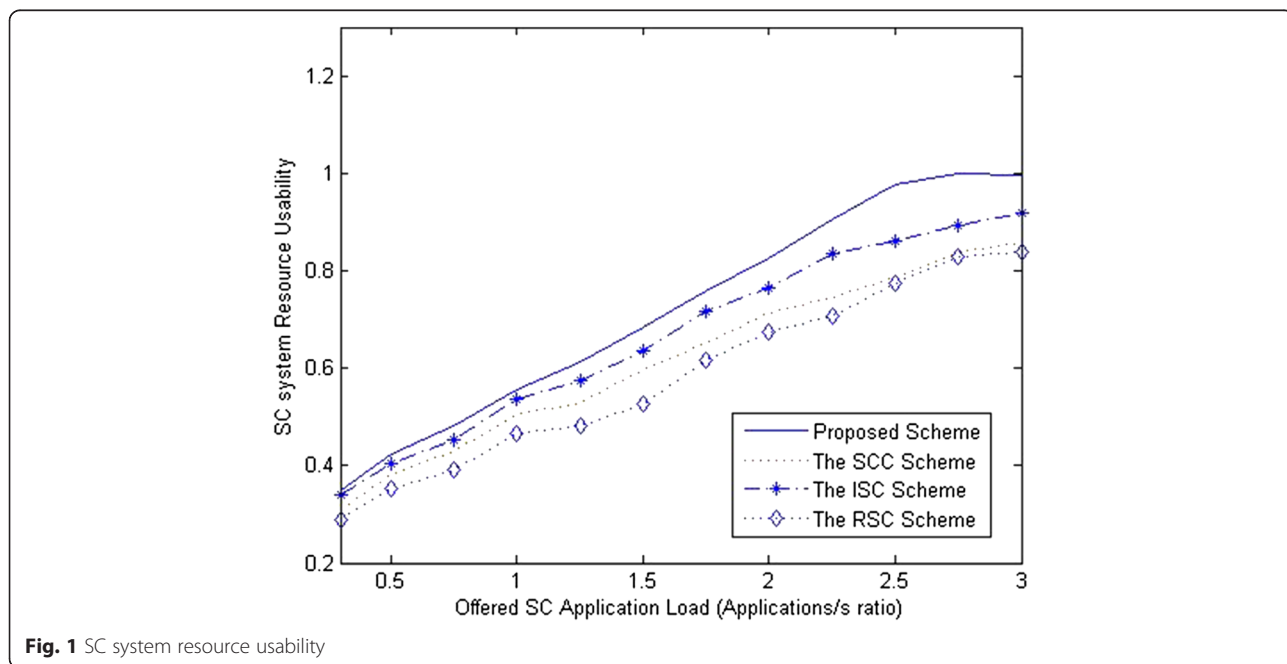


Fig. 1 SC system resource usability

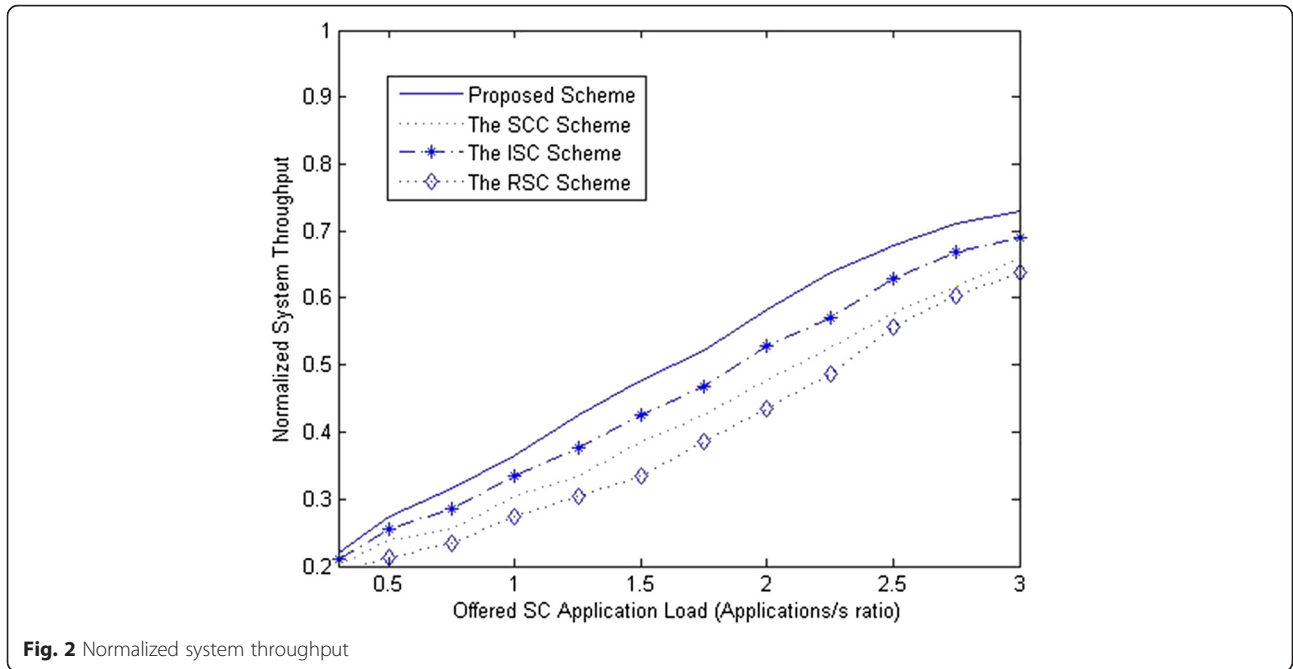
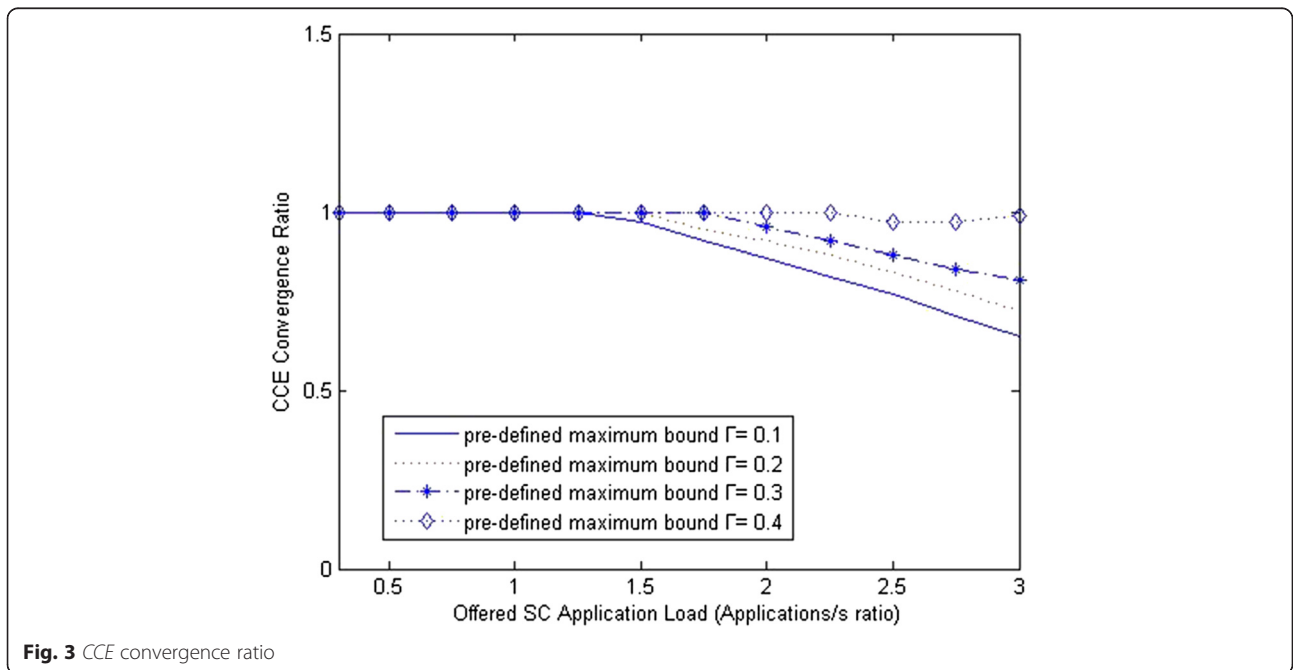


Figure 2 shows the normalized SC system throughput. From the simulation results obtained, it is observed that the proposed scheme can adapt to the current SC system situation and demonstrates better throughput owing to the iterative TS game approach. In general, excellent network throughput is a highly desirable property for real-world SC system operations. Under different application service loads, the proposed scheme can provide a higher throughput than the other schemes.

The curves in Fig. 3 illustrate the *CCE* convergence ratio under different pre-defined maximum bounds (Γ_A). If all users' relative contribution differences are within the Γ_A , *CCE* is obtained, i.e., *CCE* convergence ratio is 1. Owing to the feedback-based reciprocal mechanism, the proposed scheme can interactively learn the best altruistic propensity strategy and adaptively adjust their ϕ levels to reach the *CCE* status. Due to the self-regarding feature, these control decisions are made by an entirely distributed fashion.



As expected, *CCE* convergence ratio decreases while increasing service loads or decreasing Γ_A values.

The simulation results presented in Figs. 1 and 2 demonstrate the performance of the proposed and other existing schemes and verify that the proposed scheme can provide attractive network performance. The proposed scheme constantly monitors the current conditions for an adaptive SC system management and successfully exhibits excellent performance. As expected, the performance enhancements provided by the proposed scheme outperformed the existing schemes [10, 12, 13]. The curves shown in Fig. 3 demonstrate the *CCE* convergence ratio of our proposed scheme. This result is intuitively correct.

4 Conclusions

The ever increasing use of social networks and arrival of new computing paradigms like cloud computing has urged the need to integrate these platforms for the better and inexpensive usage of resources. Sharing cloud resources in such environments would be very helpful. This article addresses a new resource control algorithm for SC systems. Using the TS game model, users iteratively observed the received payoffs and repeatedly modified their altruistic propensities to effectively manage SC resources. The proposed scheme enables the sharing of SC resources between users via reciprocal cooperative relationships and can effectively approach the *CCE* status using a step-by-step feedback process. Compared with the existing schemes, the simulation results confirmed that the proposed game-based method could improve the performance under dynamically changing SC system environments whereas other existing schemes could not offer such an attractive performance. For future research, the SC paradigm presents a rich environment. One major area of future work is adapting the SC protocols that facilitate big data streaming from SC to Internet of Things (IoT). In addition, we are also looking further at QoS and QoE, as well as data security, privacy, and reliability issues. In parallel to these efforts, we plan to deploy the social storage cloud to provide a platform for further experimentation. In particular, we aim to explore system performance and user interactions on a much larger scale.

Competing interests

The author declares that he has no competing interests.

Acknowledgements

This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2015-H8501-15-1018) supervised by the IITP (Institute for Information & communications Technology Promotion) and was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2015R1D1A1A01060835).

Received: 22 October 2015 Accepted: 1 February 2016

Published online: 13 February 2016

References

1. M Hemmati, N Sadati, M Nili, *Towards a bounded-rationality model of multi-agent social learning in games*. IEEE ISDA'2010, 2010, pp. 142–148
2. I Jang, D Pyeon, S Kim, H Yoon, *A survey on communication protocols for wireless sensor networks*. JCSE 7(4), 231–241 (2013)
3. Z Shengbing, *Influence of relationship strengths to network structures in social network*. IEEE ISCI'2014, 2014, pp. 279–283
4. D Ardagna, M Ciavotta, M Passacantando, *Generalized Nash equilibria for the service provisioning problem in multi-cloud systems*. accepted in IEEE Transactions on Services Computing. 6(4), 429–422 (2013)
5. K Lee, I Shin, *User mobility model based computation offloading decision for mobile cloud*. JCSE 9(3), 155–162 (2015)
6. Y Liu, Y Sun, J Ryoo, S Rizvi, AV Vasilakos, *A survey of security and privacy challenges in cloud computing: solutions and future directions*. JCSE 9(3), 119–133 (2015)
7. C Xu, *Decentralized computation offloading game for mobile cloud computing*. IEEE Trans Parallel Distrib Syst 26(4), 974–983 (2015)
8. K Chard, K Bubendorfer, S Caton, OF Rana, *Social cloud computing: a vision for socially motivated resource sharing*. Services computing. IEEE Trans Serv Comput 5(4), 551–563 (2012)
9. K Chard, S Caton, O Rana, K Bubendorfer, *Social cloud: cloud computing in social networks*. IEEE CLOUD'2010, 2010, pp. 99–106. In cloud computing
10. S Caton, C Haas, K Chard, K Bubendorfer, OF Rana, *A social compute cloud: allocating and sharing infrastructure resources via social networks*. IEEE Trans Serv Comput 7(3), 359–372 (2014)
11. Kim Sungwook, *Game Theory Applications in Network Design*. IGI Global. Pennsylvania, (2014)
12. Y Zhang, M van der Schaar, *Incentive provision and job allocation in social cloud systems*. IEEE J Sel Areas Commun 31(9), 607–617 (2013)
13. Y Wang, C Guo, T Li, Q Xu, *Secure two-party computation in social cloud based on reputation*. IEEE 9th WAINA, 2015, pp. 242–245
14. K Lee, A Lippman, AS Pentland, P Maes, *People's just-in-time decisions: computational social platform to guide people's just-in-time decisions*. IEEE iThings/CPSCoM, 2013, pp. 1259–1266
15. Z Ali, RU Rasool, P Bloodsworth, *Social networking for sharing cloud resources*. IEEE Cloud and Green Computing, 2012, pp. 160–166
16. W Yilei, G Chao, L Tao, X Qiuliang, *Secure two-party computation in social cloud based on reputation*. IEEE WAINA, 2015, pp. 242–245
17. P Gajane, T Urvoy, F Clérot, *A relative exponential weighing algorithm for adversarial utility-based dueling bandits*. Proceedings of the 32nd International Conference on Machine Learning, 2015, pp. 218–227
18. S Kim, *Intervient Stackelberg game based bandwidth allocation scheme for hierarchical wireless networks*. KSII Trans Internet Inf Syst 8(12), 4293–4304 (2014)

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com